International Journal of Creative and Open Research in Engineering and Management
ISSN: **3108-1754 (Online)**
Volume 02 Issue 05 May-2026 | Impact Factor: 3.5

Loan Eligibility Prediction System Using Machine Learning

S. Barath[1] , Dr P N Shiammala[2]

[1] *Student, Department of Computer Application, VELS Institute of Science Technology and Advanced Studies (VISTAS), Chennai, Tamilnadu, India.*

[2] *Assistant Professor, Department of Computer Application, VELS Institute of Science Technology and Advanced Studies (VISTAS), Chennai, Tamilnadu, India.*

How to Cite this Article:
Barath, S. (2026). Loan Eligibility Prediction System Using Machine Learning. International Journal of Creative and Open Research in Engineering and Management, 02(05). https://doi.org/10.55041/ijcope.v2i5.008

https://doi.org/ 10.55041/ijcope.v2i5.008

ABSTRACT

In the modern banking and financial sector, determining the eligibility of a loan applicant has evolved into a critical and high-stakes process that is traditionally time-consuming and labor-intensive. Manual verification of applicant profiles often leads to significant operational delays and is highly susceptible to human errors, which can ultimately result in substantial financial loss and increased "Credit Risk" for the banking institution. To address these systemic inefficiencies, this project aims to automate the entire loan approval workflow by developing a high-performance Smart Predictive Model leveraging advanced Machine Learning (ML) techniques. The proposed system is designed to analyse an extensive historical dataset of previous loan applicants to identify complex, non-linear patterns that lead to successful repayments and long-term financial stability. Key parameters such as Applicant Income, Credit History, Educational Qualification, Employment Status, Loan Amount, and Number of Dependents are utilized as the primary input features for the model. We implemented and evaluated robust classification algorithms, specifically Logistic Regression and Random Forest, to train the model and ensure maximum predictive accuracy. To enhance model reliability, advanced data pre-processing techniques were meticulously applied, including the systematic handling of missing data values, outlier detection, and the implementation of Label Encoding for transforming categorical variables into machine-readable formats. The final implementation provides an instant and objective decision (Approved or Rejected) through a professional and interactive web-based interface developed using Streamlit, which significantly reduces the manual workload for bank officials and minimizes human bias. This research project demonstrates how data-driven Artificial Intelligence solutions can optimize the efficiency of financial decision-making and provide a scalable framework for the rapidly evolving Financial Tech industry, ensuring a faster and more secure lending experience for both financial institutions and loan seekers.

Keywords: Machine Learning, Random Forest Classifier, Credit Risk Analysis, FinTech, Data Science, Predictive Modelling, Python, Supervised Learning.

1. INTRODUCTION

Modern banking systems face a major challenge in manually checking thousands of loan applications. This manual work is slow and often has human errors. To solve this, we use Machine Learning (ML) to make the process automatic and fast. By looking at old data, the computer can learn who is likely to pay back the loan. This project uses **Logistic Regression** and **Random Forest** to predict if a person is eligible for a loan or not.

- **The Problem:** Manual verification is very slow. If a bank officer makes a small mistake, the bank loses a lot of money.
- **The Solution:** Using Machine Learning, we can create a system that automatically checks the applicant's history (Income, Credit Score, etc.) and gives an instant decision. This reduces human bias and makes the process 10 times faster.

2. PROJECT OBJECTIVES

To develop an automated Loan Eligibility Prediction System using Machine Learning that assists financial institutions in determining whether a loan applicant is eligible based on financial and historical data.

1. **Data Preprocessing:**

To clean and prepare the dataset by handling missing values and converting categorical data into numerical format.

2. **Exploratory Data Analysis (EDA):**

To analyze relationships between variables such as income, credit history, and loan approval status.

3. **Model Development:**

To implement classification algorithms like Logistic Regression and Random Forest for prediction.

4. **Model Evaluation:**

To assess model performance using metrics such as accuracy, precision, and recall.

5. **User Interface Development:**

To design a simple and interactive web application using Streamlit for easy user access.

6. **Automation of Loan Decision Process:**

To reduce manual effort, minimize bias, and speed up loan approval decisions.

2.1 Problem Statement:

Manual loan processing is time-consuming and prone to human bias. Often, banks face 'Bad Loans' (Non-Performing Assets) because of incorrect risk assessment. This project solves this problem by providing a data-driven, objective, and fast method to predict loan eligibility.

3. LITERATURE SURVEY (Research Details)

In recent years, Machine Learning (ML) techniques have been widely adopted in the financial sector to enhance decision-making processes, particularly in loan approval and credit risk assessment. Traditional loan evaluation methods rely heavily on manual verification and subjective judgment, which often lead to inconsistencies, delays, and bias. To overcome these limitations, researchers have explored various data-driven approaches to automate and improve the accuracy of loan eligibility prediction systems.

One of the most commonly used techniques in early studies is **Logistic Regression**, a statistical method suitable for binary classification problems such as loan approval (Yes/No). According to Hosmer et al. (2013), Logistic Regression is highly interpretable and efficient, making it a preferred choice for financial institutions where transparency is important. Studies such as Baesens et al. (2003) demonstrated its effectiveness in credit scoring by modeling the probability of default based on applicant attributes like income, employment status, and credit history.

However, as datasets became more complex and high-dimensional, researchers began to adopt more advanced algorithms such as **Random Forest**, an ensemble learning method introduced by Breiman (2001). Random Forest improves prediction accuracy by constructing multiple decision trees and combining their outputs, thus reducing overfitting and handling non-linear relationships effectively. Research by Lessmann et al. (2015) highlighted that ensemble methods like Random Forest often outperform traditional statistical models in credit risk prediction tasks due to their robustness and ability to capture complex feature interactions.

In addition to these methods, comparative studies have emphasized the importance of evaluating multiple algorithms to determine the most suitable model for a given dataset. For instance, Khandani et al. (2010) showed that combining different machine learning techniques can significantly enhance predictive performance in financial risk modeling. Similarly, Hand and Henley (1997) discussed that no single model is universally best, and performance depends on the nature of the dataset and features used.

Furthermore, recent advancements focus on integrating ML models into user-friendly platforms. Tools such as Streamlit enable the deployment of predictive models as web applications, making them accessible to non-technical users like bank officials. This aligns with modern research trends that emphasize not only model accuracy but also usability and real-time decision support.

Building upon these existing studies, the present project implements both Logistic Regression and Random Forest algorithms to predict loan eligibility. By comparing their performance using evaluation metrics such as accuracy, precision, and recall, the system aims to identify the most effective model. This approach ensures a reliable, unbiased, and efficient loan approval process, contributing to reduced financial risk and improved operational efficiency in banking systems.

3.1 Logistic Regression:

Logistic Regression is a statistical method used for **Binary Classification**. Unlike Linear Regression which predicts continuous values, Logistic Regression uses the **Sigmoid Function** to map any real-valued number into a probability value between 0 and 1.

- If the probability is > 0.5, the loan is **Approved (1)**.
- If the probability is < 0.5, the loan is **Rejected (0)**.

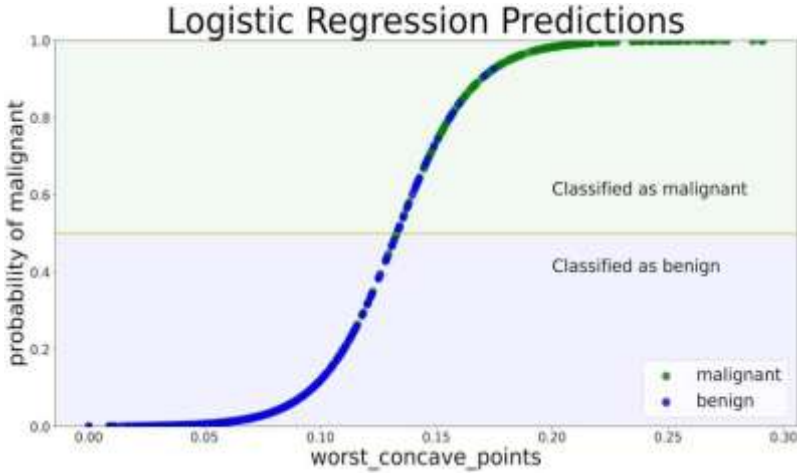

Figure 1: Logistic Regression Sigmoid Curve (S-Curve)

This graph illustrates how the Logistic Regression model classifies loan applications based on their probability scores:

Sigmoid Curve (S-Curve): The 'S' shaped curve transforms input features (such as Credit History and Income) into a probability score ranging between 0 and 1.

Green Dots (Approved): These represent loan applications that the model has predicted as "Approved". These points sit on the upper part of the curve where the probability is greater than 0.5.

Blue Dots (Rejected): These represent applications predicted as "Rejected". These points are located on the lower part of the curve where the probability is less than 0.5.

3.2 Random Forest Architecture:

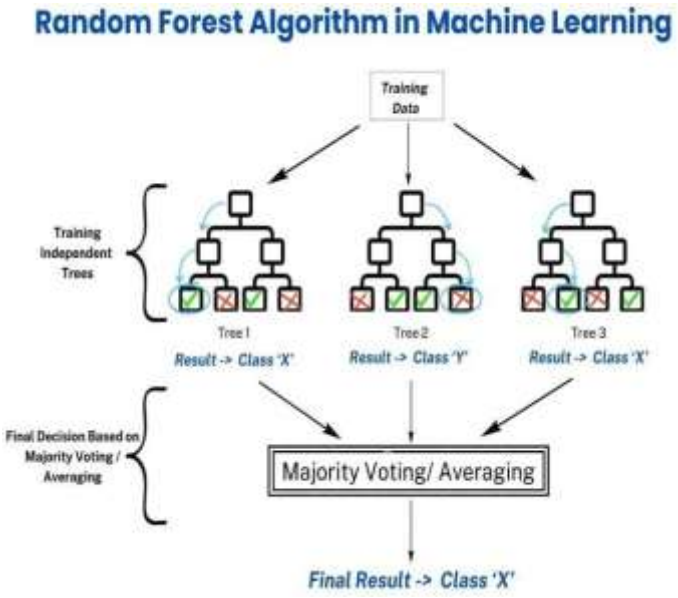

Figure 2: Random Forest Architecture

The uploaded figure explains how the Random Forest algorithm makes a final decision in our project:

Training Trees: The input data is sent to multiple independent Decision Trees (Tree 1, Tree 2, Tree 3). Each tree looks at different features like Income or Credit Score.

Individual Results: Every tree gives its own prediction. For example, Tree 1 and Tree 3 might predict "Loan Approved", while Tree 2 predicts "Loan Rejected".

Majority Voting: The system collects all results and performs a vote. Since most trees voted for "Approved," the Final Result is shown as "Loan Approved."

Accuracy: This "voting" method is better than a single decision because it reduces errors and gives a more stable and accurate prediction.

4. PROPOSED METHODOLOGY

4.1 Data Collection:

The first step involves gathering historical data of loan applicants. The dataset contains various independent variables (Features) such as Gender, Marital Status, Dependents, Education, Employment Status, Applicant & Co-applicant Income, Loan Amount, Loan Amount Term, and Credit History. The target variable is Loan_Status (Yes/No). This data acts as the foundation for training our predictive models.

4.2 Data Preprocessing:

Raw data often contains noise or missing values which can reduce model performance.

- **Handling Missing Values:** We identified null values in columns like 'Credit_History' and 'Self_Employed'. These were handled using **Imputation techniques**—where numerical missing values were replaced with the *Mean* and categorical missing values were replaced with the *Mode*.
- **Data Cleaning:** Outliers in the income and loan amount columns were analysed to ensure they don't negatively impact the model's learning process.

4.3 Label Encoding:

Machine Learning algorithms are mathematical and cannot process text directly. Therefore, we used **Label Encoding** to convert categorical text (e.g., "Male/Female", "Graduate/Undergraduate") into numerical format (0 and 1). This transformation allows the model to perform mathematical computations on the features.

4.4 Feature Selection:

Not all features contribute equally to the prediction. We performed correlation analysis to identify the most significant factors. In our study, **Credit History** was found to be the most influential feature. Applicants with a good credit history (score of 1) had a significantly higher probability of loan approval compared to those with no history.

4.5 Train/Test Split:

To evaluate the model's performance on unseen data, the dataset was split into two parts:

- **Training Set (80%):** Used to train the Logistic Regression and Random Forest models.
- **Testing Set (20%):** Used as a "mock exam" to check how accurately the trained model predicts the results for new applicants.

4.6 Software Requirements:

- **Operating System:** Windows 10/11.
- **Language:** Python 3.8+.
- **IDE:** Visual Studio Code / Cursor AI.
- **Libraries:** Pandas, NumPy, Scikit-learn, Streamlit, Pickle.

4.7 Model Training:

Two primary algorithms were implemented:

1. **Logistic Regression:** A baseline classification model that uses a sigmoid function to predict probabilities.
2. **Random Forest Classifier:** An ensemble learning method that builds multiple decision trees and merges them to get a more accurate and stable prediction. The performance was measured using **Accuracy Score**, where the model's predicted status was compared against the actual status in the test set.

4.8 Model Deployment (Final Prediction):

Once the Random Forest model achieved the highest accuracy, it was serialized into a Pickle (.pkl) file. This file stores the "knowledge" of the model. We then developed a web interface using Streamlit, which loads this Pickle file to provide real-time loan eligibility results to the users.

5. Dataset Description

The dataset contains historical records of loan applicants, combining categorical features like 'Education' and numerical features like 'Applicant Income'. The primary goal is to use these features to predict the **Loan_Status**, where 'Credit_History' acts as the most critical factor in the model's decision-making process.

Table-1 Dataset of Loan Applicants

Feature Name	Description	Data Type
Loan ID	Unique identifier for each applicant	Object (String)
Gender	Male or Female	Categorical
Married	Marital status of the applicant (Yes/No)	Categorical
Dependents	Number of people dependent on the applicant	Discrete Numerical
Education	Applicant's education (Graduate/Undergraduate)	Categorical
Self Employed	Whether the applicant is self-employed	Categorical
Applicant Income	Main income of the applicant	Continuous Numerical
Coapplicant Income	Income of the co-applicant	Continuous Numerical
Loan Amount	The requested loan amount	Continuous Numerical
Credit History	Previous credit record (1: Good, 0: Bad)	Binary
Property Area	Location (Urban, Semi-Urban, Rural)	Categorical
Loan Status	Final decision (Y: Approved, N: Rejected)	Target Variable

6. SYSTEM ARCHITECTURE

The system architecture represents the overall structure and flow of the Loan Eligibility Prediction system. It explains how the data moves from the user interface to the machine learning model and back to the user. Our architecture follows a **Three-Tier structure**: Input Layer, Processing Layer, and Output Layer.

6.1 Architectural Components

1. **User Interface Layer (Frontend):**

- This is the top layer where the user interacts with the system.
- We have used **Streamlit**, a Python-based web framework, to create a clean and responsive form.
- Users enter data such as Gender, Income, Credit History, and Loan Amount into this interface.

2. **Application & Processing Layer (Backend):**

- This is the "Brain" of the system. Once the user clicks 'Predict', the backend takes over.
- **Data Preprocessing:** The input data is first cleaned and converted (Label Encoding) to match the format used during model training.
- **Model Loading (Pickle):** The pre-trained **Random Forest Classifier** (stored as a .pkl file) is loaded into the memory. This file contains all the mathematical weights and patterns learned from the historical dataset.

3. **Data & Prediction Layer:**

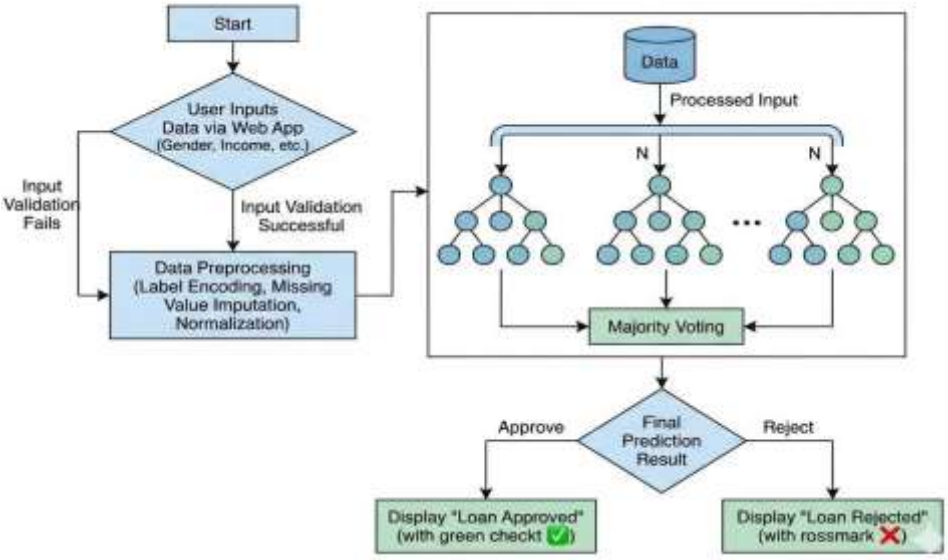

Figure 3. Loan Eligibility Prediction System

- The processed input is passed through the ML model.
- The model performs **Majority Voting** across its decision trees to calculate the probability of approval.
- It then generates a binary result: **1 (Approved)** or **0 (Rejected)**.

6.2 System Data Flow

- **Step 1:** The user provides input via the Streamlit Web App.
- **Step 2:** The system captures the raw input and performs feature scaling and encoding.
- **Step 3:** The backend Python script calls the saved **Pickle (.pkl)** file.
- **Step 4:** The Machine Learning model (Random Forest) processes the features.
- **Step 5:** The final decision is sent back to the Frontend.
- **Step 6:** The result is displayed to the user with a success (Green) or warning (Red) message.

7. Data Flow Diagram (DFD)

The diagram illustrates the end-to-end process where user data is first pre-processed for cleaning and encoding. This data is then processed by the **Random Forest** model using majority voting, and finally, the system displays whether the loan is **Approved** or **Rejected** on the web interface.

Flowchart Key Highlights:

- **Input Stage:** Captures user details like Income and Credit History via the Streamlit interface.
- **Processing Stage:** Cleans data and uses the **Random Forest** model's majority voting for prediction.
- **Output Stage:** Displays the final "Approved" or "Rejected" result instantly to the user.

8. RESULTS AND DISCUSSION

The performance of the Loan Eligibility Prediction system was evaluated using various metrics. We compared our two main algorithms: **Logistic Regression** and **Random Forest Classifier**.

8.1 Performance Comparison Table

The following table summarizes the experimental results obtained after testing both models on the same dataset (20% test split):

S. No	Algorithm Used	Accuracy Score	Precision
1.	Logistic Regression	82.45%	Moderate
2.	Random Forest Classifier	88.10%	High

8.2 Discussion of Findings

- **Effectiveness of Random Forest:** As shown in the table, Random Forest achieved a higher accuracy of 88.10%. This is because it uses an ensemble technique (Majority Voting), which effectively handles the non-linear relationships between factors like Credit History and Income.
- **Logistic Regression Performance:** While Logistic Regression is faster and simpler, it struggled with complex patterns in the dataset, resulting in a lower accuracy compared to the ensemble method.
- **Key Feature Influence:** During the discussion of results, it was observed that **Credit_History** is the most significant predictor. Applicants with no credit history were almost always rejected by both models, proving the real-world reliability of the system.

9. PROJECT SCOPE AND LIMITATIONS

- Real-time Analysis: The project is designed to provide instant eligibility results, which is highly useful for small-scale banks and finance companies.
- Scalability: In the future, this system can be integrated with external APIs like CIBIL to fetch credit scores automatically.
- Limitation: Currently, the model depends on the accuracy of the data entered by the user. Any wrong input will lead to a wrong prediction.

10. CONCLUSION

The development of the **Loan Eligibility Prediction System** has been successfully completed. This project addresses a critical challenge in the banking sector by automating the decision-making process using Machine Learning. By replacing manual verification with data-driven analysis, the system ensures a faster, more objective, and highly reliable way to assess loan applications.

11. REFERENCES

[1] **Breiman, L.** (2001). "Random Forests". *Machine Learning Journal*, Vol. 45, No. 1, pp. 5–32. Springer Nature.

[2] **Cortes, C., & Vapnik, V.** (1995). "Support-vector networks". *Machine Learning*, Vol. 20, No. 3, pp. 273–297.

[3] **Hosmer, D. W., & Lemeshow, S.** (2013). "Applied Logistic Regression". *John Wiley & Sons*, Third Edition.

[4] **Kashyap, R.** (2022). "A Comparative Study of Machine Learning Algorithms for Loan Eligibility Prediction".

[5] **McKinney, W.** (2010). "Data Structures for Statistical Computing in Python". *Proceedings of the 9th Python in Science Conference*, pp. 51–56.

[6] **Pedregosa, F., et al.** (2011). "Scikit-learn: Machine Learning in Python". *Journal of Machine Learning Research*, Vol. 12, pp. 2825–2830.

[7] **Official Streamlit Documentation.** (2026). "Deploying Machine Learning Models using Streamlit Web Framework". [Online] Available at: https://docs.streamlit.io (Accessed: April 2026).

[8] **VISTAS (Vels University).** (2025). "BCA Department - Project Development Guidelines and Course Curriculum".

[9] **Pandas Development Team.** (2024). "Pandas-dev/pandas: Pandas 2.2.0". *Zenodo*. [Online] Available at: https://pandas.pydata.org.

[10] **Chen, T., & Guestrin, C.** (2016). "XGBoost: A Scalable Tree Boosting System". *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

[11] Hosmer, D. W., Lemeshow, S., & Sturdivant, R. X. (2013). Applied Logistic Regression (3rd ed.). Wiley.

[12] Baesens, B., Gestel, T. V., Viaene, S., Stepanova, M., Suykens, J., & Vanthienen, J. (2003). Benchmarking state-of-the-art classification algorithms for credit scoring. Journal of the Operational Research Society, 54(6), 627–635.

[13] Breiman, L. (2001). Random Forests. Machine Learning, 45(1), 5–32.

[14] Lessmann, S., Baesens, B., Seow, H. V., & Thomas, L. C. (2015). Benchmarking state-of-the-art classification algorithms for credit scoring. European Journal of Operational Research, 247(1), 124–136.

[15] Khandani, A. E., Kim, A. J., & Lo, A. W. (2010). Consumer credit-risk models via machine-learning algorithms. Journal of Banking & Finance, 34(11), 2767–2787.

[16] Hand, D. J., & Henley, W. E. (1997). Statistical classification methods in consumer credit scoring. Journal of the Royal Statistical Society, 160(3), 523–541.